CENTRE
DE RECHERCHE
SUR LES TRANSPORTS
(CRT)
CENTRE
FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

MÉMOIRE D'ÉTUDIANT

C.P. 6128, SUCCURSALE A
MONTRÉAL (QUÉBEC)
CANADA
H3C 3J7

TÉLÉPHONE : (514) 343-7575
TÉLEX : 05-24146 BIBPOLYTEC
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-648

GÉNÉRATEUR D'ITINÉRAIRES D'ENLÈVEMENT DE LA NEIGE

par

Johanne Gilbert

Mémoire présenté à la Faculté des études supérieures en vue de l'obtention du grade de Maître ès sciences (M.Sc.) en informatique. L'auteur jouissait d'une bourse du Conseil de recherches en sciences naturelles et en génie du Canada et d'une bourse dans le cadre de la subvention Action structurante du Centre de recherche sur les transports.

Université de Montréal

Centre de recherche sur les transports - Publication # 648
Laboratoire d'optimisation du DIRO - Publication # LO 89-7
Département d'Informatique
et de Recherche opérationnelle - Publication # 695

Avril 1989

90-01404/

Dépôt légal - 4e trimestre 1989

SOMMAIRE

Plusieurs problèmes de recouvrement d'arcs ont déjà été traités, notamment les problèmes de balayage de rues et de la collecte des ordures ménagères. Dans ce mémoire, nous nous intéressons plus particulièrement au problème d'enlèvement de la neige dans un secteur urbain.

Notre modélisation a donné lieu à un problème de grande taille en programmation mathématique en nombres entiers avec contraintes très spécifiques. Nous avons choisi de le résoudre par une méthode heuristique. Celle-ci utilise deux variantes d'une méthode d'insertion, soit l'insertion simple et l'insertion en parallèle.

MOTS-CLÉS : DÉNEIGEMENT, TOURNÉES ET HORAIRES, RÉSEAUX ORIENTÉS, PLUS COURTS CHEMINS, COMMIS VOYAGEUR, OPTIMISATION, HEURISTIQUES.

TABLE DES MATIERES

Introduction 1

CHAPITRE 1 Définition du problème 3

CHAPITRE 2 Modélisation 11

2.1 Notation 13

2.2 Variables de décision 15

2.3 Objectif 15

2.4 Contraintes 16

2.5 Modèle 19

CHAPITRE 3 Approche de résolution 23

3.1 Méthode d'insertion 23

3.1.1 Calcul du détour 24

3.2 Variantes de la méthode d'insertion 25

3.2.1 Insertion simple 26

4.3.2 Germe pour chacune des étapes 55

4.3.3 Pourcentage de remplissage des étapes permises 56

4.3.4 Choix de l'étape la moins remplie en pourcentage 57

4.3.5 Ajustements 57

CHAPITRE 5 Utilisation et adaptation de fonctions existantes 58

5.1 Editeur de réseau 58

5.2 Base de tronçons 61

5.3 Grande table de distances entre les tâches 61

5.4 Petite table de distances entre les tronçons 63

5.5 Voyageur de commerce 64

CHAPITRE 6 Application à un exemple réel et résultats 67

6.1 Test avec les données d'une tempête type 69

6.2 Difficulté d'avoir plus de données 71

6.3 Améliorations et extensions 71

Conclusion 73

Bibliographie 75

LISTE DES FIGURES

3.1 Calcul du détour 24

3.2 Côtés de rues utilisés 28

6.1 Réseau 68

INTRODUCTION

Plusieurs problèmes de recouvrement d'arcs ont déjà été traités tels que la collecte des ordures ménagères, le balayage de rues, le postier chinois, les itinéraires d'autobus d'écoliers en milieu rural, etc. Nous analysons un autre problème de ce type, soit l'enlèvement de la neige dans un milieu urbain. Il s'agit pour l'équipe de chargement de passer sur toutes les rues et d'y enlever la neige de chaque côté. Certaines contraintes sont très spécifiques à ce contexte. Ainsi, certaines rues doivent être traitées dans les premières journées et l'on ne peut pas déneiger les deux côtés d'une même rue au cours de la même demi-journée.

Notre modélisation nous a conduit à une formulation en terme d'un problème de recouvrement de noeuds (que représentent les tâches de déneiger les divers segments de rues) plutôt qu'en un problème de recouvrement d'arcs comme on aurait pu s'y attendre de prime abord. Ceci a donné lieu à un problème de grande taille en programmation mathématique en nombres entiers très difficilement traitable avec une méthode exacte. Nous avons donc développé une heuristique qui tout en respectant les diverses contraintes arrive à générer les itinéraires

d'enlèvement de la neige. Cette heuristique consiste à utiliser simultanément deux variantes d'une méthode d'insertion qui a pour objet d'insérer chaque noeud (tâche) au cours d'une journée de déneigement. La première variante dénotée insertion simple est particulièrement utile pour insérer les tâches prioritaires au cours des premières journées. Pour sa part, l'insertion en parallèle (seconde variante) sert particulièrement à bien équilibrer la charge de travail au cours des dernières journées.

Dans le premier chapitre, on donne une description détaillée du problème considéré. Dans le second chapitre, on le modélise en un problème de grande taille en programmation mathématique non-linéaire avec variables 0-1. Les variables, les contraintes et l'objectif y sont énoncés. Par la suite, dans le troisième chapitre, on donne l'idée générale de l'approche de résolution. On y explique ce qu'est une méthode d'insertion et on précise les détails des variantes choisies. Puis dans le quatrième chapitre, on traite de l'implantation de l'algorithme. Entre autres, on précise les regroupements des tâches et l'implantation des méthodes d'insertion simple et d'insertion en parallèle. Au cinquième chapitre, on explique comment on a utilisé et adapté à nos besoins certaines procédures déjà disponibles tels que les algorithmes de Dijkstra et du voyageur de commerce. Finalement, au sixième chapitre, on présente les résultats obtenus lors de la résolution d'un problème réel.

CHAPITRE 1

DEFINITION DU PROBLEME

Pour les fins du déneigement, la Ville de Montréal a été divisée en un certain nombre de territoires bien définis, chacun faisant l'objet d'un contrat individuel [7]. La Ville de Montréal accorde des contrats de déneigement pour une durée de cinq ans, c'est-à-dire cinq hivers. Chaque période d'hiver s'étend du 15 novembre jusqu'au 15 mars de l'année suivante.

Le travail de déneigement comprend les opérations de déblaiement, de chargement et de transport de neige. Il ne doit être effectué qu'aux endroits où la Ville est tenue de le faire et ce, dans les délais prescrits. Le déneigement doit être fait complètement sur toutes les rues visées par le contrat en enlevant toute la neige accumulée des deux côtés en bordure de chaque rue.

Dans ce mémoire, on s'intéresse à l'opération de chargement. Cette opération comprend l'enlèvement mécanique de toute la neige accumulée sur les trottoirs et la chaussée et s'il y a lieu, sur les terres-

pleins ou les mails qui les séparent. L'opération de chargement est généralement exigée pour les accumulations de plus de cinq centimètres de neige provenant d'une ou de plusieurs précipitations.

Plusieurs règles spécifiques régissent le chargement. Pour mieux préciser le problème, nous avons résumé les principales. Le début du chargement dépend des facteurs suivants:

-l'heure de la fin de la chute de neige basée sur les rapports météorologiques;

-l'heure de la pose des panneaux interdisant le stationnement;

-le début de la période régulière de travail sachant que la journée normale de travail est divisée en deux périodes régulières, celle de jour débutant à 7h et celle de nuit débutant à 19h.

Tout en respectant les directives données concernant l'heure de la pose des panneaux, l'entrepreneur doit commencer le travail de chargement le plus tôt possible :

-si la fin de la chute de neige survient entre 7h et 19h, le chargement commence au plus tard à 7h le lendemain matin;

-si la fin de la chute de neige survient entre 19h d'une journée et 7h le lendemain, le chargement commence au plus tard à 19h cette deuxième journée.

Le chargement doit être entièrement terminé dans les délais suivants :

a) pour une précipitation de moins de 20 cm, 96 heures après la fin de la chute de neige;

b) pour une précipitation de 20 cm à 25 cm, 108 heures après la fin de la chute de neige;

c) pour une précipitation de 25 cm ou plus, 120 heures après la fin de la chute de neige, à moins qu'un prolongement ne soit accordé.

Sauf durant le déblaiement, l'entrepreneur doit indiquer au moyen de panneaux appropriés, les endroits où il a l'intention d'effectuer son travail de déneigement. Ces panneaux interdisent le stationnement durant des périodes définies, soit durant les deux périodes de travail, et ils doivent être mis en place à l'intérieur de délais acceptables pour les tiers. Ces panneaux lui permettent de déplacer, si nécessaire, les véhicules des tiers nuisant à ses opérations.

En plus de se conformer aux règles précédentes, l'entrepreneur est tenu de suivre les directives suivantes:

a) à moins d'exception, il n'est pas permis d'interdire le stationnement sur les deux côtés d'un même segment de rue aux mêmes heures;

b) il est interdit de charger la neige aux heures de pointe sur les rues où il y a des enseignes d'interdiction de stationnement durant ces heures sur un côté ou l'autre de la rue;

c) l'équipe de chargement est tenue de suivre les règles du code de la route, notamment les interdictions de virage et les sens uniques, et on cherche à éviter de faire des virages en U;

d) sur les artères principales à circulation intense et où la largeur de la chaussée est supérieure à la moyenne des autres rues, l'entrepreneur verra à dégager ces artères et charger la neige le plus rapidement possible car elles sont considérées comme prioritaires.

Chaque segment de rue à déneiger peut être vu comme une tâche à exécuter. Le problème devient alors d'agencer ces tâches dans le but de réduire les déplacements à vide tout en tenant compte des segments prioritaires et des autres contraintes. L'approche de résolution retenue en est une d'insertion. On se sert d'une approche d'insertion simple pour les premières journées, là où il y a des tâches prioritaires, tandis qu'une approche d'insertion en parallèle est utilisée pour les journées suivantes constituées uniquement de tâches secondaires.

Pour vérifier l'efficacité de l'approche proposée dans ce mémoire, nous avons analysé un territoire particulier du nord de Montréal où une seule équipe de chargement est utilisée. Ce territoire illustré à la figure 6.1, est délimité par les rues Lajeunesse (excl.), Jarry (incl.), Foucher

(incl.), Crémazie (excl.), St-Hubert (incl.), De Liège (incl.), Boyer (incl.), Crémazie (excl.), Marquette (incl.), Jean-Talon (incl.). Au chapitre 6, nous donnerons plus de détails relatifs à ce problème spécifique.

D'autres problèmes de recouvrement d'arcs ont déjà été traités, entre autres les problèmes de balayage de rues, de la collecte des ordures ménagères, des postiers, des itinéraires d'autobus d'écoliers en milieu rural, etc. On décrit sommairement dans ce qui suit quelques unes des approches pour traiter ces problèmes où en général plusieurs véhicules sont utilisés.

Dans [1] et [3], on traite d'un problème de recouvrement d'arcs où il s'agit de construire les itinéraires pour le balayage de rues en minimisant la distance totale d'allège. C'est une adaptation du problème du postier chinois sur un réseau avec arcs orientés. Puisque plusieurs équipements sont utilisés sur plusieurs périodes de travail, deux grandes approches s'offrent pour traiter le problème:

i) "Cluster first, Route second"

Pour chacune des périodes considérées, on identifie les tâches qui peuvent être exécutées et on les répartit entre les véhicules. Ensuite il s'agit pour chacun de ceux-ci d'établir un itinéraire parmi les tâches qui lui ont été assignées.

ii) "Route first, Cluster second"

Pour chacune des périodes considérées, il s'agit d'établir un grand itinéraire incluant toutes les tâches qui peuvent être exécutées et ensuite de le briser en plusieurs itinéraires admissibles pour chacun des véhicules.

Comme on désire éviter le chevauchement entre les itinéraires d'une même période et que l'objectif n'est pas uniquement de minimiser la distance totale d'allège, on favorise la première approche dans [1] et [3].

Les journées sont divisées en périodes de temps basées sur les interdictions de stationnement. Pour chaque période de temps, on indentifie d'abord les rues pouvant être balayées. En répartissant celles-ci entre les divers véhicules, on génère pour chaque véhicule particulier un sous-graphe pouvant comporter une ou plusieurs composantes connexes. Chaque composante est traitée individuellement afin d'y équilibrer le nombre d'arcs qui entrent et qui sortent de chaque noeud en résolvant un problème de transport. Le réseau augmenté des arcs d'allège constitue un graphe orienté d'Euler.

Pour arriver à spécifier l'ordre suivant lequel les arcs sont balayés, on utilise un problème d'affectation qui pénalise à différents niveaux divers types de virages tel que présenté dans [10,12]. L'opération peut engendrer plusieurs sous-cycles dans une même composante. Pour former un cycle incorporant tous les arcs, il suffit d'utiliser les points communs entre les divers sous-cycles. Pour identifier l'itinéraire d'un

véhicule pour la période considérée, il suffit de combiner les cycles de ses différentes composantes connexes en solutionnant un problème de recouvrement d'arbre minimal. La cédule quotidienne d'un balai de rue est alors constituée d'une suite d'itinéraires construits pour chacune des périodes de la journée.

Le problème de la collecte des ordures ménagères se formule lui aussi comme un problème de postier chinois [9]. On analyse le cas où chacun des arcs du réseau doit être desservi et où un seul véhicule est utilisé. En général, on doit repasser sur certains arcs déjà desservis pour compléter le service sur l'ensemble du réseau, et ces déplacements correspondent au temps d'allège. Tout comme pour le problème de balayage de rues, nous devons d'abord générer un graphe orienté d'Euler en ajoutant des arcs d'allège pour équilibrer le nombre d'arcs entrant et sortant à chaque noeud. Nous devons ensuite résoudre un problème d'affectation et réunir l'ensemble des sous-cycles de l'unique composante du réseau en un tour d'Euler.

Dans [8], on a développé des heuristiques pour réunir l'ensemble des sous-cycles d'une composante en un tour d'Euler. L'objectif de la méthode est de réduire le nombre total de virages et de réduire ou d'éliminer les virages à gauche. L'heuristique consiste en un ensemble de règles et de configurations types d'itinéraires qu'on doit chercher à reproduire le plus possible. Cette approche est destinée aux réseaux qui ne sont pas des graphes orientés d'Euler et génère souvent des

itinéraires qui sont plus longs que l'itinéraire optimal. Cependant, cette approche peut être adaptée aux graphes orientés d'Euler et donne de très bons résultats.

Dans [5], un problème de recouvrement d'arcs avec capacité sur les véhicules est traité. Il s'agit d'établir le parcours des autobus d'écoliers en milieu rural. L'objectif est de réduire les détours tout en équilibrant le nombre d'écoliers sur les parcours. On utilise alors une méthode d'insertion en parallèle selon deux stratégies:

1) Etant donné un arc, déterminer le parcours existant où l'arc pourrait être inséré de telle sorte que le détour encouru soit minimal.

2) Etant donné un parcours, déterminer lequel des arcs qui restent devrait y être inséré.

CHAPITRE 2

MODELE MATHEMATIQUE

Nous pouvons formuler notre modèle comme un problème de recouvrement d'arcs de type "Postier chinois non-symétrique" [2] où chaque arc du réseau correspond à un segment de rue à déneiger et est orienté dans le sens où le chargement peut se faire. Chaque arc correspondant à un segment de rue entre deux intersections est caractérisé par sa longueur qui est fixe et par un temps de chargement qui dépend de la largeur de la rue, de son mobilier urbain ainsi que du nombre de centimètres de neige tombée. Puisque le temps de chargement est fixe, l'optimisation porte sur la distance "allège" parcourue entre chacun des arcs à déneiger. Il est à noter qu'il est équivalent de minimiser le temps ou la distance d'allège étant donné que la vitesse à vide est constante.

Chaque arc à déneiger peut être vu comme une tâche et le problème peut être reformulé en terme de tâches à exécuter séquentiellement où l'objectif est de minimiser l'allège entre les tâches. La distance d'allège entre chaque couple de tâches est égale à la

distance entre la fin de la première tâche et le début de la seconde. Ainsi, notre modèle originalement défini en terme de recouvrement d'arcs se prête à une reformulation en terme de recouvrement de noeuds (les tâches) sans avoir à tripler le nombre de noeuds comme c'est le cas dans [11]. Ce nouveau problème peut être comparé à un problème de voyageur de commerce multiple non-symétrique [4] mais avec contraintes assez spéciales.

Pour notre application, chaque journée commence et finit au dépôt représenté par un noeud d'indice 0. Rappelons que lorsqu'on fait appel à la procédure, le nombre de journées est déjà spécifié. De plus, notons que chaque journée est séparée en deux périodes, chacune comportant elle-même un nombre spécifique d'étapes.

Si l'étape "e" de la journée "d" n'est pas la dernière de cette journée, alors il faut considérer le temps d'allège pour se rendre de la dernière tâche de "e" à la première tâche de l'étape suivante, et cette allège est comptabilisée à l'étape suivante. Ceci est nécessaire pour s'assurer qu'à l'étape suivante, on débute à l'endroit où s'est terminée l'étape "e". Pour faire les bons chaînages entre les tâches d'une étape à l'autre d'une même journée, on se sert de tâches bidons dans notre modèle. Puisque chacune des journées est divisée en "E" étapes, "E-1" tâches bidons les séparent.

Voici maintenant une description détaillée du modèle.

2.1 Notation

D : ensemble des journées pour faire le déneigement; $d \in D$

E(d) : ensemble des étapes de la journée d; $e \in E(d)$

$\bar{E}(d)$: ensemble des étapes de la journée d à l'exclusion de la première

$E_k(d)$: partition de E(d), $1 \leq k \leq K$

(i.e. $E_1(d) \cup E_2(d) \cup ... \cup E_K(d) = E(d)$

$E_k(d) \cap E_l(d) = \emptyset \quad 1 \leq k,l \leq K, \quad k \neq l$)

e^- : étape précédant e la même journée (e^- défini pour $e \in \bar{E}(d)$)

J : ensemble des tâches; $j \in J$

$\bar{j}$: tâche correspondant au déneigement de l'autre côté du segment de rue associé à la tâche j

$\bar{J}$: $= J \cup \{ 0 \}$

b : noeud bidon

$N^+(e)$ et $N^-(e)$: Des ensembles de tâches admissibles pour e défini comme suit:

$$N^+(e) = \begin{cases} \bar{J} \text{ si } e \in E(d) \text{ est la dernière étape de la journée d} \\ \bar{J} \cup \{b\} \text{ sinon} \end{cases}$$

$$N^-(e) = \begin{cases} \bar{J} \text{ si } e \in E(d) \text{ est la première étape de la journée d} \\ \bar{J} \cup \{b\} \text{ sinon} \end{cases}$$

d_{ij} : distance minimale entre la fin de la tâche i et le début de la tâche j.

t_{ij} : temps minimal entre la fin de la tâche i et le début de la tâche j, c'est-à-dire (d_{ij} / vitesse d'allège)

l_j : temps de déneigement de la tâche j

L_e : temps total disponible au cours de l'étape e

$$a_j^{ed} = \begin{cases} 1 \text{ si la tâche j peut être exécutée à l'étape e de la journée d} \\ 0 \text{ sinon} \end{cases}$$

2.2 Variables de décision

$$y_j^{ed} = \begin{cases} 1 \text{ si la tâche j est exécutée à l'étape e de la journée d} \\ 0 \text{ sinon} \end{cases}$$

$$x_{ij}^{ed} = \begin{cases} 1 \text{ si la tâche i précède la tâche j à l'étape e de la journée d} \\ 0 \text{ sinon} \end{cases}$$

2.3 Objectif

$$\text{MIN} \sum_{d \in D} \sum_{e \in E(d)} \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ij}^{ed} + \sum_{d \in D} \sum_{e \in E(d)} \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ib}^{e^- d} * x_{bj}^{ed}$$

La deuxième partie de l'objectif ajuste le calcul du temps inter-tâche entre la dernière tâche d'une étape "e" et la première tâche de l'étape suivante de la même journée, $e \in \bar{E}(d)$, via une tâche bidon.

2.4 Contraintes

Chaînage de tâches

Si une tâche est effectuée au cours d'une étape, une tâche quelconque doit la succéder et la précéder. Cependant, cette tâche peut aussi être le garage ou une tâche "bidon" utile pour le chaînage entre les étapes d'une même journée.

$$\sum_{i \in N^-(e)} x_{ij}^{ed} = y_j^{ed} \qquad e \in E(d), d \in D, j \in J$$

(une tâche précède)

$$\sum_{j \in N^+(e)} x_{ij}^{ed} = y_i^{ed} \qquad e \in E(d), d \in D, i \in J$$

(une tâche succède)

$$\sum_{i \in J} x_{i0}^{ed} = \begin{cases} 1 \text{ si } e \in E(d) \text{ est la dernière étape de } d \\ 0 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

(une tâche précède au garage)

$$\sum_{j \in J} x_{0j}^{ed} = \begin{cases} 1 \text{ si } e \in E(d) \text{ est la première étape de } d \\ 0 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

(une tâche succède au garage)

$$\sum_{i \in J} x_{ib}^{ed} = \begin{cases} 0 \text{ si } e \in E(d) \text{ est la dernière étape de } d \\ 1 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

(une tâche précède au bidon)

$$\sum_{j \in J} x_{bj}^{ed} = \begin{cases} 0 \text{ si } e \in E(d) \text{ est la première étape de } d \\ 1 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

(une tâche succède au bidon)

Priorité de la tâche

La tâche j doit être exécutée au cours des journées de l'ensemble D(j) qui dépend du nombre de centimètres de neige tombée et de la priorité de la rue.

$$\sum_{d \in D(j)} \sum_{e \in E(d)} y_j^{ed} = 1 \qquad j \in J$$

Admissibilité de la tâche

Certaines tâches prioritaires ne peuvent être complétées à certaines étapes de la journée.

$$y_j^{ed} \leq a_j^{ed} \qquad e \in E(d), d \in D, j \in J$$

Incompatibilité des deux côtés d'un arc au cours d'un même sous-ensemble d'étapes d'une journée (par exemple le sous-ensemble peut correspondre à une demi-journée)

$$\sum_{e \in E_k(d)} y_j^{ed} + y_{\bar{j}}^{ed} \leq 1 \qquad 1 \leq k \leq K, d \in D, j \in J$$

Temps limite d'une étape d'une journée

$$\sum_{j \in J} l_j * y_j^{ed} + \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ij}^{ed} \leq L_e$$

$$e \in (E(d) - \bar{E}(d)), d \in D$$

$$\sum_{j \in J} l_j * y_j^{ed} + \sum_{i \in J} \sum_{j \in J} (t_{ij} * x_{ij}^{ed} + t_{ij} * x_{ib}^{e^- d} * x_{bj}^{ed}) \leq L_e$$

$$e \in \bar{E}(d), d \in D$$

Elimination de sous-tours

Des contraintes d'élimination de sous-tours au cours d'une étape d'une journée sont ajoutées.

Variables Binaires

$$y_j^{ed} = 0,1 \qquad e \in E(d), d \in D, j \in J$$

$$x_{ij}^{ed} = 0,1 \qquad e \in E(d), d \in D, i,j \in \bar{J} \cup \{b\}$$

2.5 Modèle

Le modèle se résume donc comme suit:

$$\text{MIN} \sum_{d \in D} \sum_{e \in E(d)} \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ij}^{ed} + \sum_{d \in D} \sum_{e \in E(d)} \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ib}^{e^- d} * x_{bj}^{ed}$$

s.à

$$\sum_{i \in N^-(e)} x_{ij}^{ed} = y_j^{ed} \qquad e \in E(d), d \in D, j \in J$$

$$\sum_{j \in N^+(e)} x_{ij}^{ed} = y_i^{ed} \qquad e \in E(d), d \in D, i \in J$$

$$\sum_{i \in J} x_{i0}^{ed} = \begin{cases} 1 \text{ si } e \in E(d) \text{ est la dernière étape de } d \\ 0 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

$$\sum_{j \in J} x_{0j}^{ed} = \begin{cases} 1 \text{ si } e \in E(d) \text{ est la première étape de } d \\ 0 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

$$\sum_{i \in J} x_{ib}^{ed} = \begin{cases} 0 \text{ si } e \in E(d) \text{ est la dernière étape de } d \\ 1 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

$$\sum_{j \in J} x_{bj}^{ed} = \begin{cases} 0 \text{ si } e \in E(d) \text{ est la première étape de } d \\ 1 \text{ sinon} \end{cases}$$

$e \in E(d), d \in D$

$$\sum_{d \in D(j)} \sum_{e \in E(d)} y_j^{ed} = 1 \qquad j \in J$$

$$y_j^{ed} \leq a_j^{ed} \qquad e \in E(d), d \in D, j \in J$$

$$\sum_{e \in E_k(d)} y_j^{ed} + y_j^{\bar{e}d} \leq 1 \qquad 1 \leq k \leq K, d \in D, j \in J$$

$$\sum_{j \in J} l_j * y_j^{ed} + \sum_{i \in J} \sum_{j \in J} t_{ij} * x_{ij}^{ed} \leq L_e$$

$$e \in (E(d) - \bar{E}(d)), d \in D$$

$$\sum_{j \in J} l_j * y_j^{ed} + \sum_{i \in J} \sum_{j \in J} (t_{ij} * x_{ij}^{ed} + t_{ij} * x_{ib}^{\bar{e}d} * x_{bj}^{ed}) \leq L_e$$

$$e \in \bar{E}(d), d \in D$$

Des contraintes d'élimination de sous-tours au cours d'une étape d'une journée sont ajoutées.

$$y_j^{ed} = 0,1 \qquad e \in E(d), d \in D, j \in J$$

$$x_{ij}^{ed} = 0,1 \qquad e \in E(d), d \in D, i,j \in \bar{J} \cup \{b\}$$

Le modèle proposé définit un problème de programmation non-linéaire avec des variables 0-1. L'objectif et la contrainte de temps limite sont quadratiques tandis que toutes les autres contraintes sont linéaires. Cela donne lieu à un problème avec plusieurs variables ainsi que plusieurs contraintes. En effet, si on possède "α" tâches et "β" étapes au total pour procéder au chargement, on se retrouve avec :

- $(\alpha + \alpha^2) * \beta$ variables

- $\approx (4 * \alpha * \beta + 5 * \beta + \alpha)$ contraintes en plus des contraintes d'élimination de sous-tours.

Dans l'exemple qui sera analysé au chapitre 6, il y a 470 tâches et le déneigement s'échelonne sur quatre journées totalisant douze étapes de travail. Ainsi, le nombre de variables et de contraintes est très impressionnant. On voit facilement qu'un problème modeste peut devenir rapidement un problème de grande taille. Ceci nous a motivé à développer une approche heuristique pour traiter ce problème.

CHAPITRE 3

APPROCHE DE RESOLUTION

3.1 Méthode d'insertion

Comme le problème posé est de grande taille et qu'il comporte plusieurs contraintes telles la limite de temps de chaque étape, la priorité de certaines tâches et la non-admissibilité de certaines tâches à certaines périodes, nous favorisons une méthode heuristique plutôt qu'une méthode exacte. Nous proposons également comme approche de résolution, une méthode d'insertion.

Pour les problèmes de recouvrement de noeuds [2], une procédure d'insertion est un processus itératif insérant successivement chaque noeud dans le tour de recouvrement. Etant donné un sous-tour comportant certains noeuds, la procédure doit d'abord déterminer lequel des noeuds (qui n'est pas encore dans le sous-tour) devrait être le prochain à être inséré au sous-tour (étape de sélection) et puis détermine l'endroit dans le sous-tour où il devrait être inséré (étape d'insertion) de façon à produire le plus petit détour possible.

3.1.1 Calcul du détour

Le détour est calculé selon la stratégie de "Cheapest Insertion" [2]. En effet, supposons que l'on veuille insérer la tâche (T) entre les tâches (P) et (S) agissant respectivement comme tâches précédente et suivante, le détour passe de "d_{PS}" à "$d_{PT} + d_{TS} - d_{PS}$".

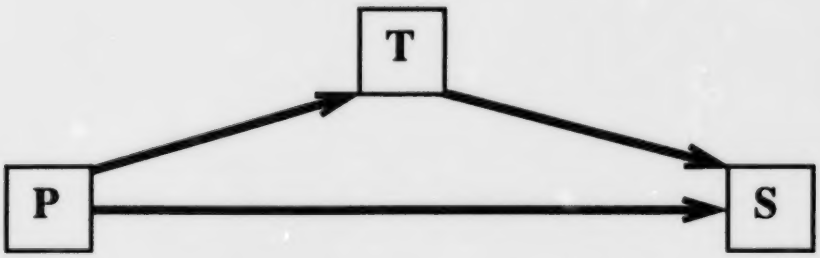

Figure 3.1 Calcul du détour

Pour notre problème, le détour peut prendre différentes formes selon que la tâche à insérer est ou n'est pas la première de l'étape et selon l'endroit où elle est insérée dans celle-ci.

Dans le cas où ce n'est pas la première tâche à être choisie pour une étape, il est toujours possible de calculer le détour occasionné. Quelque soit l'endroit où nous voulons placer une tâche, elle en suit une autre et en précède une autre. La tâche qui précède (P) peut faire partie de l'étape courante, être la dernière tâche de l'étape précédente de la même journée, ou lorsqu'aucun de ces deux cas ne se présente, être le

garage. De la même façon, la tâche qui succède (S) peut faire partie de l'étape courante, être la première tâche de l'étape suivante de la même journée, ou lorsqu'aucun de ces deux cas ne se présente, être le garage.

Considérons maintenant le cas où la tâche est la première pour une étape. S'il existe au moins une tâche dans l'étape précédente ou l'étape suivante de la même journée, il est possible de calculer le détour. Le détour occasionné est alors la somme de la distance entre la dernière tâche de l'étape précédente (si elle existe) et la tâche sélectionnée pour l'étape courante et de la distance entre la tâche sélectionnée pour l'étape courante et la première tâche de l'étape suivante (si elle existe). Par contre, s'il n'y a pas d'étapes adjacentes de la même journée comportant déjà une tâche, on ne peut calculer le détour comme précédemment et alors on doit opter pour un autre critère. On a retenu le critère de la longueur pour sélectionner la tâche la plus longue en mètres. Ce critère est arbitraire mais a été retenu car on a constaté qu'il devient plus difficile de placer une longue tâche à mesure que l'étape se remplie.

3.2 Variantes de la méthode d'insertion

Plusieurs variantes de la méthode d'insertion (cheapest insertion, nearest insertion, arbitrary insertion, furthest insertion) [2] ont déjà été analysées. Pour résoudre le problème qui nous concerne, nous utilisons deux variantes de la méthode d'insertion soit l'insertion simple et l'insertion en parallèle [5].

3.2.1 Insertion simple

Dans la méthode d'insertion simple, nous comblons successivement chaque étape de travail. Ainsi en commençant par la première étape, il s'agit de choisir une à une les tâches de façon à minimiser le détour occasionné selon les formules précédentes. Lorsqu'il ne reste plus suffisamment de temps dans l'étape pour placer une autre tâche ou qu'il ne reste plus de tâches qui peuvent être choisies à cette étape, on passe à l'étape suivante, et ainsi de suite tant qu'il reste des tâches à placer. Les itinéraires des premières étapes générés par cette méthode sont de meilleure qualité car plusieurs tâches peuvent être choisies. Par contre, lorsque nous sommes rendus aux dernières étapes, le choix des tâches est de plus en plus limité et les itinéraires générés sont de moins bonne qualité. Nous favorisons tout de même l'utilisation de cette approche pour les premières étapes où la solution doit absolument être de qualité supérieure pour déneiger rapidement les segments prioritaires.

3.2.2 Insertion en parallèle

Ce qui distingue la méthode d'insertion en parallèle, c'est qu'on n'attend pas nécessairement qu'une étape soit complétée avant de passer à la suivante. Nous commençons par mettre une tâche (germe) dans chacune des étapes lequel germe est choisi d'après le critère de la longueur. Puis, nous choisissons l'étape qui est la moins remplie en pourcentage et nous déterminons alors la tâche qui peut y être insérée et qui minimise le détour occasionné. On passe ainsi d'une étape à l'autre

et nous augmentons nos chances d'avoir des étapes mieux équilibrées. Il est possible également que l'on ne veuille pas remplir complètement certaines étapes en insertion en parallèle. Lorsqu'on a déjà utilisé un certain pourcentage du temps maximal disponible pour l'étape, on peut alors décider de ne plus choisir cette étape. Cette méthode est utilisée lorsque l'on désire équilibrer la qualité relative de certaines étapes.

3.3 Liste des tâches éligibles pour une étape donnée

Que ce soit pour la variante d'insertion simple ou d'insertion en parallèle, lorsqu'on a déterminé laquelle des étapes est la prochaine à recevoir une nouvelle tâche, il s'agit de choisir une tâche parmi l'ensemble des tâches qui sont éligibles à cette étape.

Comme la souffleuse doit toujours être à droite des camions lors du chargement, ceci influence la sélection des tâches éligibles pour les segments de rues à sens unique. On ne peut déneiger qu'un seul côté au cours d'une période et il faut faire en sorte de charger la neige du côté droit du sens unique d'abord. Pour les segments de rue secondaire à double sens, les deux côtés de rue sont éligibles mais en pratique, on ne permettra que de faire un choix parmi ceux-ci. Si c'est un segment de rue prioritaire à double sens, il faut voir à déneiger le côté de rue qui sera le plus utilisé aux prochaines heures de pointe. Ce côté de rue constitue un choix de "premier ordre" dans le cas des tâches prioritaires.

A titre d'illustration, dans le secteur considéré, ce sont les directions ouest et sud qui sont utilisées à l'heure de pointe matinale tandis que les directions est et nord le sont à l'heure de pointe de fin d'après-midi. C'est donc dire que les côtés nord et ouest sont les plus utilisés le matin et les côtés sud et est en fin d'après-midi.

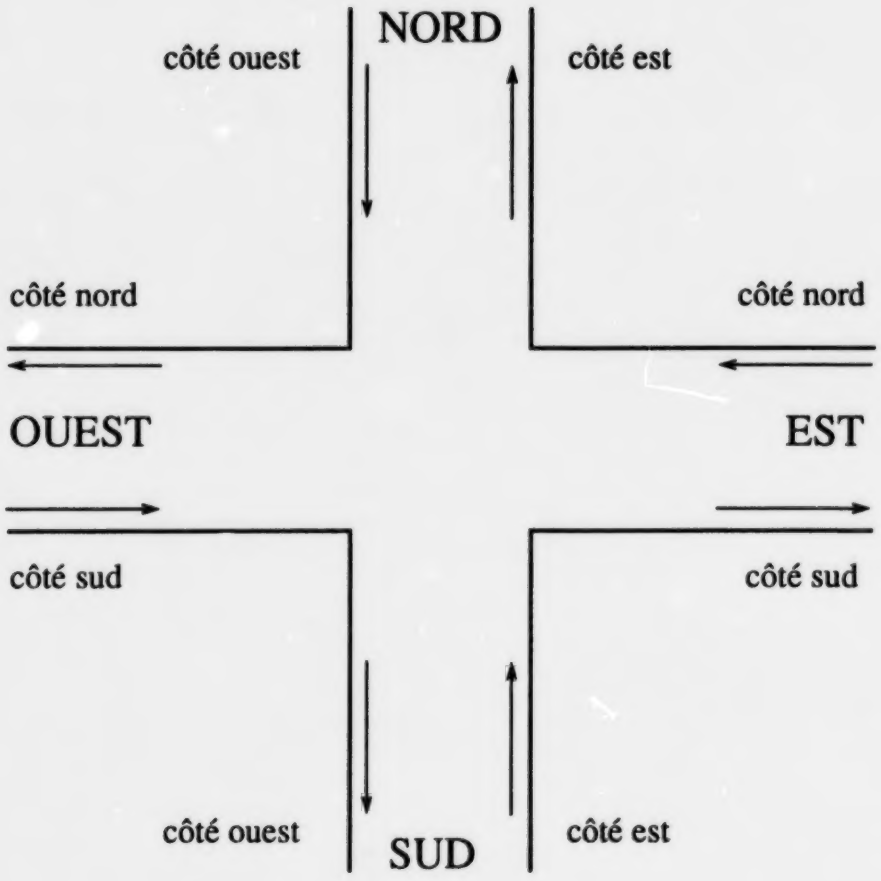

Figure 3.2 Côtés de rues utilisés

Pour les rues prioritaires à double sens, bien qu'un seul côté bien spécifique peut être assigné à une période déterminée, il arrive de permettre de sélectionner l'autre côté au cours de cette période à titre de choix de "second ordre". Cela se produit lors d'une étape en dehors des heures de pointe où toutes les tâches prioritaires éligibles de premier ordre ont déjà été choisies. On permet alors de déneiger ces côtés de rues qui ne sont pas assignées à cette période à la condition que l'autre côté de ceux-ci ne soient pas déneigés à cette même période. Ceci est permis pour accélérer le déneigement des rues prioritaires.

3.4 Voyageur de commerce

Malgré le fait que les tâches sont insérées au meilleur endroit dans l'étape pour minimiser le détour occasionné, il y a peut-être moyen de changer l'ordre de la séquence de tâches à exécuter dans l'étape de façon à minimiser l'allège total. Pour y parvenir, nous faisons appel à une adaptation de l'algorithme du voyageur de commerce [4]. Etant donné un ensemble de tâches à exécuter et une table de distances entre chacune de ces tâches, on trouve la meilleure façon d'ordonnancer ces tâches de façon à minimiser l'allège total. Pour des problèmes de taille réduite, la solution est exacte sinon elle est approximative. En général, on fait appel à cet algorithme lorsqu'une étape est comblée à capacité ou lorsqu'il n'y a plus de tâches éligibles disponibles pour l'étape.

3.5 Détails des méthodes d'insertion choisies

Voici maintenant les détails des méthodes d'insertion simple et d'insertion en parallèle.

3.5.1 Insertion simple

.De la première à la dernière étape permise

..Faire une liste des tâches éligibles pour l'étape courante

..Tant que la liste de tâches éligibles n'est pas vide et qu'il reste suffisamment de temps dans l'étape courante pour placer une autre tâche

...Si la liste de tâches déjà sélectionnées est vide

Choisir comme première tâche celle qui minimise le détour occasionné ou qui est la plus longue selon le cas

...Sinon

Choisir comme prochaine tâche celle qui entraîne le plus petit détour

...S'il a été possible de sélectionner une nouvelle tâche respectant les diverses contraintes

Faire les mises-à-jour

Si la liste de tâches éligibles de l'étape est devenue vide

On fait appel au voyageur de commerce

Si on traite présentement des tâches prioritaires

On fait une liste de tâches éligibles de second ordre

...Sinon (aucune tâche n'a été sélectionnée)

S'il y a eu au moins une insertion dans l'étape depuis le dernier appel au voyageur de commerce

On fait appel au voyageur de commerce

Sinon

Il ne reste plus suffisamment de temps dans l'étape pour placer une autre tâche

3.5.2 Insertion en parallèle

•Identifier les étapes permises pour l'insertion en parallèle

•Choisir une première tâche pour toutes les étapes permises et faire les mises-à-jour qui s'imposent

•Tant qu'il reste des tâches et qu'on n'a pas dépassé le pourcentage de remplissage des étapes permises

..Choisir l'étape permise la moins remplie en pourcentage

..Faire une liste des tâches éligibles pour cette étape

..Faire le choix d'une prochaine tâche pour cette étape

...S'il a été possible de choisir un tâche respectant les diverses contraintes

Faire les mises-à-jour

Si la liste de tâches éligibles est devenue vide

On fait appel au voyageur de commerce

...Sinon

S'il y a eu au moins une insertion dans l'étape depuis le dernier appel au voyageur de commerce

On fait appel au voyageur de commerce

Sinon

Cette étape n'est plus permise pour l'insertion en parallèle

3.6 Ordre d'appel des méthodes d'insertion choisies

Nous avons décrit précédemment les méthodes d'insertion simple et d'insertion en parallèle. Nous utilisons ces deux variantes des méthodes d'insertion lors d'une même exécution du programme.

Tout d'abord, on commence par placer les tâches prioritaires en insertion simple au cours des premières étapes afin d'avoir les plus beaux itinéraires possibles pour celles-ci. Ces tâches prioritaires ne peuvent être placées que lors des étapes ne faisant pas partie des heures de pointe. Nous complétons ces premières étapes avec des tâches secondaires en utilisant l'insertion simple. Pour les tâches secondaires qui restent, on complète les étapes ultérieures avec l'insertion simple ou l'insertion en parallèle.

Les grandes lignes de l'heuristique sont donc les suivantes:

1) Placer les tâches prioritaires en insertion simple au cours des premières étapes

2) Compléter ces premières étapes avec des tâches secondaires en utilisant l'insertion simple

3) Pour les étapes ultérieures ne comportant que les tâches secondaires qui restent, on a le choix:

i) Utiliser l'insertion simple

ii) Utiliser l'insertion en parallèle

a) Placer les tâches secondaires en insertion en parallèle jusqu'à un certain remplissage des étapes

b) Placer les tâches secondaires qui restent en insertion simple

CHAPITRE 4

IMPLANTATION

4.1 Formation des tronçons

Même si on résolvait avec une méthode exacte le modèle présenté au chapitre 2, les itinéraires pourraient comporter beaucoup de virages et donc beaucoup de zigzags. Ainsi, ce genre d'itinéraire serait très différent de ceux utilisés actuellement et ne serait pas acceptable à cause des difficultés d'opérations qui en résulteraient.

Il est de beaucoup préférable de rechercher des itinéraires semblables à ceux utilisés en pratique. Ceci nous amène à faire du regroupement de tâches, c'est-à-dire des "tronçons" de un à plusieurs arcs correspondant à des segments consécutifs d'un même côté de rue. Dans le modèle précédent, les tronçons remplacent les tâches. Ceci a pour effet de réduire le nombre de virages dans la solution obtenue.

Ceci ne veut pas dire pour autant que tous les arcs d'un même côté de rue appartiennent au même tronçon, en général. Une rue peut d'abord être séparée en plusieurs tronçons pour des raisons physiques

(par exemple lorsqu'un parc la sépare en deux parties distinctes) ou pour des raisons de circulation (par exemple un changement de direction du sens unique). Egalement, pour des raisons subjectives, on peut séparer une rue en plusieurs tronçons de façon à refléter la façon de faire du contracteur.

En procédant ainsi, on réduit considérablement la taille du problème et on améliore les chances d'obtenir une solution acceptable. Pour l'exemple traité, nous sommes passés de 470 tâches à 122 tronçons. Malgré le fait que la taille du problème ait diminué en introduisant la notion de tronçon, il n'en reste pas moins que c'est un problème non-linéaire qui demeure difficile à résoudre. Par abus de langage, on désigne par tronçon à sens unique, un tronçon faisant partie d'un bout de rue à sens unique et on désigne par tronçon à double sens, un tronçon faisant partie d'un bout de rue à double sens.

4.2 Insertion simple

4.2.1 Etapes permises en insertion simple

Tel que mentionné précédemment, on a un ensemble de journées composées chacune d'un certain nombre d'étapes pour procéder au déneigement. La première étape de travail permise pour procéder au déneigement de l'ensemble du territoire peut tout aussi bien faire partie d'une période de jour que d'une période de nuit selon l'heure de la fin de la chute de neige. De plus, il faut se rappeler que dépendant du nombre

de centimètres de neige tombée, il y a un temps limite pour procéder au déneigement et de là découle une dernière étape "e_{finale}" pour procéder au déneigement de l'ensemble du territoire.

Pour les tronçons prioritaires, la dernière étape permise est l'étape "e_{finale}" bien qu'en pratique, on tente toujours de les placer en premier lieu. Soit "p_{prio}" la denière période utilisée pour placer les tronçons prioritaires, "d_{prio}" la journée dont cette période fait partie et "e_{prio}" la dernière étape de cette période. En général, on finit de placer les tronçons prioritaires dans les deux ou trois premières journées. D'autre part, il faut noter que les tronçons prioritaires ne peuvent être placés au cours des étapes qui ont lieu pendant les heures de pointe matinale ou d'après-midi.

La dernière étape permise pour placer les tronçons secondaires en insertion simple est l'étape "e_{finale}" si on ne permet pas l'insertion en parallèle sinon c'est l'étape "e_{prio}".

Voici maintenant pour une journée type les étapes permises et défendues pour placer les différents tronçons en insertion simple

i) L'étape ayant lieu pendant l'heure de pointe matinale n'est pas accessible aux prioritaires mais permise pour les secondaires.

ii) L'étape ayant lieu le jour en dehors des heures de pointe est permise pour tous les types de tronçons.

iii) L'étape ayant lieu pendant l'heure de pointe de fin d'après-midi n'est pas accessible aux prioritaires mais permise pour les secondaires.

iv) L'étape ayant lieu durant la soirée ou la nuit est permise pour tous les types de tronçons.

Au départ, on doit séparer les périodes de jour et de nuit en un certain nombre d'étapes pour éviter de placer des tronçons prioritaires aux heures de pointe. Comme après l'étape "e_{prio}", il n'y a plus de tronçons prioritaires, toutes les étapes faisant partie d'une même période peuvent être regroupées en une seule dont la durée est égale à la somme des durées des étapes la composant. La combinaison des étapes est complétée que l'on permette l'insertion en parallèle ou non.

A chaque fois qu'on ajoute un tronçon dans une étape, cela signifie qu'on occupe une partie de la durée de cette étape. Le temps de parcourir le tronçon constitue le principal élément, et à cela vient s'ajouter le temps d'allège supplémentaire occasionné par l'insertion de ce tronçon dans l'étape concernée. De plus, il faut se rappeler que le temps d'allège entre les tronçons de deux étapes successives est comptabilisé à la seconde étape.

On considère qu'il reste suffisamment de temps dans l'étape pour placer un autre tronçon si on respecte l'une des conditions suivantes:

i) Si on n'a pas encore sélectionné aucun tronçon pour l'étape.

ii) Si à la dernière tentative d'insertion d'un tronçon pour cette étape, on a réussi à sélectionner un tronçon ou on a fait appel au voyageur de commerce.

4.2.2 Faire une liste des tronçons éligibles pour une étape donnée

Voici donc les détails de la procédure pour former la liste des tronçons éligibles pour une étape donnée. On donne d'abord deux versions pour les tronçons prioritaires puis une autre version pour les tronçons secondaires.

<u>i) Tronçons prioritaires éligibles de premier ordre</u>

•Du premier au dernier tronçon prioritaire

••S'il a déjà été déneigé à une autre étape

ce n'est pas un tronçon éligible

••Sinon s'il est à sens unique

Si on traite une étape en dehors des heures de pointe

Si c'est le côté droit

c'est un tronçon éligible

Sinon (c'est le côté gauche)

si on a déjà choisi le côté droit pour cette étape

ce n'est pas un tronçon éligible

Sinon

c'est un tronçon éligible

Sinon (i.e., si on traite une étape comprenant les heures de pointe de matin ou de fin d'après-midi)

ce n'est pas un tronçon éligible

..Sinon (il est à double sens)

Si on traite une étape en dehors des heures de pointe

Si le côté du tronçon est le plus utilisé aux prochaines heures de pointe

c'est un tronçon éligible

Sinon

ce n'est pas un tronçon éligible

Sinon (i.e., si on traite une étape comprenant les heures de pointe de matin ou de fin d'après-midi)

ce n'est pas un tronçon éligible

ii) Tronçons prioritaires éligibles de second ordre

•Du premier au dernier tronçon prioritaire

••S'il a déjà été déneigé à une autre étape

ce n'est pas un tronçon éligible

••Sinon si ce tronçon est à double sens

Si on traite une étape en dehors des heures de pointe

Si l'autre côté est déneigé au cours de la même étape

ce n'est pas un tronçon éligible

Sinon

c'est un tronçon éligible

Sinon (i.e., si on traite une étape comprenant les heures de pointe de matin ou de fin d'après-midi)

ce n'est pas un tronçon éligible

••Sinon (c'est un sens unique)

ce n'est pas un tronçon éligible

iii) Tronçons secondaires éligibles

•Du premier au dernier tronçon secondaire

••S'il a déjà été déneigé à une autre étape

ce n'est pas un tronçon éligible

••Sinon si c'est un sens unique

Si c'est le côté droit

c'est un tronçon éligible

Sinon (c'est le côté gauche)

Si on a déjà choisi le côté droit pour cette étape

ce n'est pas un tronçon éligible

Sinon

c'est un tronçon éligible

••Sinon (i.e., il est à double sens)

c'est un tronçon éligible

{Les deux côtés sont dans la liste}

4.2.3 Choix du premier tronçon d'une étape

Pour déterminer lequel des tronçons éligibles agira à titre de germe pour l'étape, il faut d'abord constater dans laquelle des situations on se trouve. Cela permettra de déterminer combien d'allège occasionnera l'insertion d'un premier tronçon dans une étape étant donné le fait qu'il est possible qu'on ait déjà des tronçons dans les étapes précédente ou suivante de la même journée. L'allège n'est calculée que pour les étapes d'une même journée et il n'y a donc pas d'allège entre la dernière étape d'une journée et la première étape de la journée suivante. Si aucune allège ne peut être occasionnée, on se basera alors sur le critère de la longueur.

En effet, si on cherche le germe parmi les tronçons prioritaires, on sait qu'il n'y aura pas d'allège car les étapes précédente ou suivante de la même journée ne sont constituées que de tronçons secondaires qui ne seront placés que plus tard. On ne peut donc pas choisir le germe "prioritaire" en fonction du détour occasionné. Il n'y a pas non plus de détour occasionné lorsqu'on fait le choix d'un germe "secondaire" en insertion simple pour la première étape d'une journée suivant "d_{prio}". Dans ces deux cas, on retiendra le critère de la longueur et on choisira le tronçon le plus long.

Si on cherche un germe parmi des tronçons secondaires, il peut y avoir quatre modes d'évaluation alors qu'il n'y en a qu'un lorsqu'on en cherche un parmi des tronçons prioritaires (car pour eux aucun détour

ne peut être occasionné). Dénotons par "étape intermédiaire", une étape de la journée "d" à l'exclusion de la première et de la dernière étape de cette journée.

Mode 0 : Pas de détour occasionné

-Si c'est la première étape de la journée d et qu'il n'y a pas de tronçon déjà sélectionné dans l'étape suivante.

-Si c'est une étape intermédiaire et qu'il n'y a pas de tronçon déjà sélectionné ni dans l'étape précédente ni dans la suivante.

-Si c'est la dernière étape de la journée d et qu'il n'y a pas de tronçon déjà sélectionné dans l'étape précédente.

Mode 1 : Détour occasionné en considérant l'étape précédente

-Si c'est une étape intermédiaire, qu'il y a au moins un tronçon déjà sélectionné dans l'étape précédente mais qu'il n'y en a pas dans l'étape suivante.

-Si c'est la dernière étape de la journée d et qu'il y a au moins un tronçon déjà sélectionné dans l'étape précédente.

Mode 2 : Détour occasionné en considérant les étapes précédente et suivante

-Si c'est une étape intermédiaire et qu'il y a au moins un tronçon déjà sélectionné dans chacune des étapes précédente et suivante.

Mode 3 : Détour occasionné en considérant l'étape suivante

-Si c'est la première étape de la journée d et qu'il y a au moins un tronçon déjà sélectionné dans l'étape suivante.

-Si c'est une étape intermédiaire, qu'il n'y a pas de tronçon déjà sélectionné dans l'étape précédente mais qu'il y en a au moins un dans l'étape suivante.

Après avoir déterminé le mode d'évaluation, il s'agit de parcourir la liste des tronçons éligibles et pour chacun d'eux déterminer la valeur du détour ou de la longueur, selon le cas, et par la suite de choisir le meilleur tronçon.

Les détails de la procédure sont donc :

•Déterminer le mode d'évaluation

•Initialiser le "Meilleur tronçon vu jusqu'à date"

Le meilleur tronçon vu jusqu'à date occasionne un détour infini et a une longueur nulle, c'est un tronçon "imaginaire".

•Du premier au dernier tronçon éligible

••Si aucun détour ne peut être occasionné

Si le tronçon courant est plus long que le meilleur vu jusqu'à date

Choisir ce tronçon comme meilleur vu jusqu'à date

..Sinon

Selon le mode d'évaluation (1-2-3), on calcule le détour occasionné

Si le détour est "meilleur", le tronçon est choisi comme meilleur vu jusqu'à date

On choisit comme germe le meilleur tronçon vu jusqu'à date à condition que celui-ci ne soit pas le tronçon "imaginaire".

4.2.4 Choix du prochain tronçon de l'étape

Afin de déterminer lequel des tronçons éligibles sera choisi pour être le prochain à être inséré, il faut d'abord déterminer dans quelle situation on se trouve. Cela permettra de déterminer combien d'allège occasionnera l'insertion d'un autre tronçon dans l'étape en tenant compte du fait qu'il peut être inséré tout aussi bien au début de l'étape, entre d'autres tronçons de l'étape ou à la fin de l'étape et qu'il peut y avoir déjà au moins un tronçon dans les étapes précédente ou suivante de la même journée. Il y a cinq modes d'évaluation possibles lorsqu'on place un tronçon suivant. En effet, le détour occasionné se présente sous cinq formes:

TRONÇON PLACE AU DEBUT DE L'ETAPE

MODE 4 Détour occasionné en considérant les étapes précédente et courante

-Si ce n'est pas la première étape de la journée, s'il y a au moins un tronçon déjà sélectionné à l'étape précédente et si on place le tronçon au début de l'étape.

MODE 5 Détour occasionné en ne considérant que l'étape courante

-Si c'est la première étape de la journée et si on place le tronçon au début de l'étape.

-Si ce n'est pas la première étape de la journée, s'il n'y a pas de tronçon déjà sélectionné à l'étape précédente et si on place le tronçon au début de l'étape.

TRONÇON PLACE ENTRE D'AUTRES TRONÇONS DE L'ETAPE

MODE 6 Détour occasionné en ne considérant que l'étape courante

-Si on place le tronçon à une position intermédiaire de l'étape, i.e. ni au début ni à la fin de l'étape.

TRONÇON PLACE A LA FIN DE L'ETAPE

MODE 7 Détour occasionné en ne considérant que l'étape courante

-Si c'est la dernière étape de la journée et si on place le tronçon à la fin de l'étape.

-Si ce n'est pas la dernière étape de la journée, s'il n'y a pas de tronçon déjà sélectionné à l'étape suivante et si on place le tronçon à la fin de l'étape.

MODE 8 Détour occasionné en considérant les étapes courante et suivante

-Si ce n'est pas la dernière étape, s'il y a au moins un tronçon déjà sélectionné à l'étape suivante et si on place le tronçon à la fin de l'étape.

Après avoir identifié les modes d'évaluation selon l'endroit où l'on peut placer un tronçon, il s'agit de parcourir la liste des tronçons éligibles et pour chacun d'eux déterminer la valeur du détour selon la position où on veut l'insérer. On choisira le tronçon qui placé à une certaine position dans l'étape donnera le meilleur résultat.

Les détails de la procédure sont donc :

•Déterminer les modes d'évaluation

•On initialise le meilleur tronçon vu jusqu'à date

Le meilleur tronçon vu jusqu'à date occasionne un détour infini, c'est un tronçon "imaginaire".

•Du premier au dernier tronçon éligible

••Si l'autre côté du tronçon n'a pas été retenu pour cette étape

•••On essaie d'insérer le tronçon au début, i.e. avant le premier tronçon de l'étape

Selon le mode (4 ou 5), on calcule le détour occasionné

Si en plaçant le tronçon courant au début de l'étape, le détour est "meilleur", il est choisi comme meilleur vu jusqu'à date

•••Du premier à l'avant-dernier tronçon déjà sélectionnés à l'étape

On essaie de placer le tronçon éligible entre ce tronçon et son successeur

Selon le mode 6, on calcule le détour occasionné

Si le détour est "meilleur", le tronçon est choisi comme meilleur vu jusqu'à date

...On essaie d'insérer le tronçon à la fin de l'étape i.e. après le dernier tronçon de l'étape

Selon le mode (7 ou 8), on calcule le détour occasionné

Si en plaçant le tronçon courant à la fin de l'étape, le détour est "meilleur", il est choisi comme meilleur vu jusqu'à date

On choisit le meilleur tronçon vu jusqu'à date à condition que celui-ci ne soit pas le tronçon "imaginaire".

4.2.5 Calcul du détour occasionné

Voici maintenant comment se fait le calcul du détour occasionné selon les différents modes d'évaluation.

Notation

A: Dernier tronçon de l'étape précédente

B: Premier tronçon de l'étape courante

C: Tronçon de l'étape courante après lequel on veut insérer un tronçon

D: Tronçon de l'étape courante avant lequel on veut insérer un tronçon

E: Dernier tronçon de l'étape courante

F: Premier tronçon de l'étape suivante

X: Tronçon que l'on désire insérer

Dénotons par Distance[G][H] le plus court chemin entre la fin du tronçon G et le début du tronçon H. Le détour occasionné en insérant le tronçon X dans l'étape est alors :

Si mode = 1

Détour = Distance[A][X]

Si mode = 2

Détour = Distance[A][X] + Distance[X][F]

Si mode = 3

Détour = Distance[X][F]

Si mode = 4

Détour = Distance[A][X] + Distance[X][B] - Distance[A][B]

Si mode = 5

Détour = Distance[X][B]

Si mode = 6

Détour = Distance[C][X] + Distance[X][D] - Distance[C][D]

Si mode = 7

Détour = Distance[E][X]

Si mode = 8

Détour = Distance[E][X] + Distance[X][F] - Distance[E][F]

4.2.6 Comparaison de la valeur d'un tronçon au meilleur vu jusqu'à date

Cette procédure détermine si un tronçon placé à un certain endroit d'une certaine étape d'une journée est préférable au meilleur tronçon rencontré jusqu'à date pour cette étape. Si c'est le cas, on compare alors la valeur des autres tronçons à celui-ci tant qu'on n'en aura pas trouvé un meilleur. Rappelons qu'en introduisant un tronçon dans une étape, le temps qui s'ajoute est le temps de déneiger le tronçon (longueur du tronçon / vitesse de déneigement du tronçon) additionné au temps d'allège (détour occasionné / vitesse d'allège).

Le détail de la procédure est donc :

•Si l'autre côté du tronçon a déjà été sélectionné pour une étape de la même période

Ce tronçon n'est pas meilleur

•Si en introduisant le tronçon, le temps maximum disponible pour l'étape est dépassé

Ce tronçon n'est pas meilleur

•Si le détour occasionné par le tronçon est supérieur au détour occasionné par le meilleur tronçon vu jusqu'à date

Ce tronçon n'est pas meilleur

•Si le détour occasionné par le tronçon est inférieur au détour occasionné par le meilleur tronçon vu jusqu'à date

Ce tronçon est meilleur

•Si le détour occasionné par le tronçon est équivalent au détour occasionné par le meilleur tronçon vu jusqu'à date, on choisit le tronçon qui occasionne le moins de changements de rues:

Si le meilleur tronçon vu jusqu'à date est situé sur la même rue que son successeur ou son prédécesseur de l'étape

Ce tronçon n'est pas meilleur

Sinon si le tronçon est sur la même rue que son successeur ou son prédécesseur de l'étape

Ce tronçon est meilleur

Sinon

Ce tronçon n'est pas meilleur

4.2.7 Mises-à-jour

Lorsqu'un tronçon est sélectionné pour faire partie d'une étape, il y a certains ajustements à faire dans les structures de données. On enlève le tronçon de la liste de tronçons éligibles pour l'étape en question et on l'insère à l'endroit approprié dans la liste de tronçons sélectionnés à cette étape. On ne permet plus que ce tronçon soit choisi à une autre étape et

on prendra note du jour et de l'étape où ce tronçon a été placé. Comme le déneigement de ce tronçon et l'allège encouru demande du temps, on augmente d'autant le temps déjà utilisé pour l'étape.

De plus, si on place le tronçon en dernière position de l'étape et que cette étape n'est pas la dernière de la journée, il faut répartir le temps d'allège entre les étapes courante et suivante comme suit afin de comptabiliser le temps d'allège inter-tâche à l'étape suivante:

MODE	ETAPE COURANTE	ETAPE SUIVANTE
2	Distance[A][X]	Distance[X][F]
3		Distance[X][F]
8	Distance[E][X]	Distance[X][F] - Distance[E][F]

4.3 Insertion en parallèle

Nous allons expliquer ici la façon dont a été implantée la procédure d'insertion en parallèle. Evidemment, certaines fonctions utilisées sont les mêmes que pour l'insertion simple.

4.3.1 Identifier les étapes permises en insertion en parallèle

La première étape permise est la première étape suivant l'étape "e_{prio}" et la dernière étape permise est l'étape "e_{finale}". De plus, les étapes sont combinées selon le processus déjà présenté car il n'y a pas de tronçons prioritaires dans ces étapes. Comme il n'y a qu'une étape par période, on se retrouve avec un maximum de deux étapes par jour.

4.3.2 Germe pour chacune des étapes

Tous les tronçons qui ne sont pas encore assignés à une étape font partie de la liste de tronçons éligibles à devenir le germe d'une étape. Il s'agit de passer d'une étape permise à une autre en assignant un germe basé sur le critère de la longueur. On permet toutefois de placer d'autres tronçons dans une étape avant même que le germe de toutes les étapes ne soit spécifié lorsque cela semble avantageux.

Voici donc la procédure.

•Tant qu'il reste une étape dont le germe n'a pas été spécifié

••Tant qu'on n'a pas sélectionné l'étape qui recevra un germe et qu'on n'a pas analysé toutes les étapes une à la suite de l'autre

•••Si l'étape courante est vide

On trouve le tronçon le plus long dans la liste de tronçons éligibles pour les germes

Si possible, on calcule le détour occasionné selon le mode (1-3) et cette étape est sélectionnée si on respecte la contrainte de temps

...Sinon elle contient au moins un tronçon

Cette étape est sélectionnée si le tronçon engendrant le détour minimal (calculé comme en 4.2.4 selon l'un des modes 4-5-6-7-8) a une valeur inférieure à un certain seuil.

...Si l'étape est sélectionnée

Faire les mises-à-jour qui s'imposent

4.3.3 Pourcentage de remplissage des étapes permises

Comme l'éxécution du programme est rapide, il y a moyen de faire une exécution ne permettant pas l'insertion en parallèle. Le résultat donne en général des étapes remplies presqu'à capacité à l'exception de la dernière étape. On peut alors faire une autre exécution du programme, cette fois-ci en permettant l'insertion en parallèle dans l'espoir d'obtenir de meilleurs résultats. En se basant sur le pourcentage de remplissage de la dernière étape obtenu en faisant de l'insertion simple (P_1), on va déterminer un pourcentage de remplissage de la dernière étape (P_2) qui ne devra pas être excédé au cours de l'insertion en parallèle. Le pourcentage (P_2) est inférieur au pourcentage (P_1) car il n'inclue pas le dernier tronçon qui peut être inséré.

Il est aussi possible que l'on ne soit pas intéressé à placer tous les tronçons en insertion en parallèle. Peut- être qu'on désire remplir les étapes à une certaine capacité et compléter le tout en insertion simple. On se définira alors un autre pourcentage (P_3) applicable à toutes les étapes sauf la dernière. C'est donc dire qu'avant de choisir une étape afin d'y insérer un tronçon, le pourcentage de remplissage de l'étape ne devra pas dépasser (P_2) ou (P_3) selon le cas. Si $P_2 = P_3 = 1.0$, c'est qu'on permet de remplir toutes les étapes à capacité.

4.3.4 Choix de l'étape la moins remplie en pourcentage

Ce critère nous permet d'être plus équitable envers toutes les étapes. Pour chacune des étapes permises, on fait le quotient du temps déjà utilisé sur le temps maximal permis et on choisit l'étape qui a le quotient le moins élevé.

(temps déjà utilisé) / (temps maximal * (P_2 ou P_3))

4.3.5 Ajustements à faire pour les sens uniques

En fonctionnant en parallèle, on n'est pas en mesure de contrôler le fait qu'on doit d'abord déneiger le côté droit d'un tronçon à sens unique puisqu'on ne s'occupe plus de remplir entièrement les étapes une à la suite de l'autre comme en insertion simple. C'est pourquoi lorsqu'on a fini de placer tous les tronçons, il s'agit de faire les ajustements nécessaires lorsque la contrainte n'est pas respectée.

CHAPITRE 5

UTILISATION ET ADAPTATION DE FONCTIONS EXISTANTES

5.1 Editeur de réseau

Pour réaliser ce projet, nous avons utilisé un réseau de type HOTOCARS (HOraires et Tournées sur Ordinateurs, Computer Aided Routing and Scheduling) disponible au Centre de recherche sur les transports. Tel que mentionné dans [6], la structure de données résultante comporte beaucoup d'informations, notamment plusieurs liens qui facilitent les opérations requises par les programmes travaillant avec le réseau : programmes de localisation, d'édition du réseau, de confection d'une table de distances, de production de cheminements exacts, etc.

L'information décrivant un réseau routier de type HOTOCARS se rattache à l'une des entités suivantes :

noeud : point dans un espace à deux dimensions;

arc : lien établissant une connexion entre deux noeuds;

rue : ensemble d'arcs formant le plus souvent un chemin contigu entre deux noeuds extrêmes;

région : ensemble des noeuds, arcs et rues couvrant un territoire donné, c'est un sous-ensemble d'un réseau.

Les noeuds et les arcs sont des entités physiques, c'est-à-dire décrivant des objets existants, les rues sont des regroupements d'arcs auxquels on donne une identification pour en faciliter le repérage, enfin, les régions sont des entités fonctionnelles qui ont été créées pour limiter la quantité d'informations à manipuler lorsqu'un traitement ne s'applique qu'à une portion du réseau. De plus, il est bon de noter que chacune des rues du réseau se voit assigner un sens, c'est le sens du réseau. C'est donc dire que chacun des arcs membre de l'une de ces rues a un noeud origine et un noeud destination associés. Pour notre application particulière, les rues "Nord Sud" sont orientées vers le nord tandis que les rues "Est Ouest" sont orientées vers l'est.

Dans le cadre du projet HOTOCARS de confection de routes et horaires, on dispose de beaucoup d'information dont plusieurs listes facilitant l'accès aux noeuds, arcs et rues. Cette structure permet de réaliser efficacement les opérations suivantes:

.Tracer une carte du réseau : ce qui implique de connaître les arcs et les coordonnées géographiques des noeuds;

.Localiser sur le réseau par adresse ou par le nom des rues formant une intersection : ce qui implique de connaître les noms des rues, les numéros civiques aux coins des rues et les rues qui s'entrecoupent;

.Trouver un plus court chemin : ce qui implique de connaître les arcs qui sont adjacents à un noeud donné, les voies à sens unique, les interdictions de virage, etc.;

.Modifier le réseau : ce qui implique un accès rapide aux différentes informations pour les éditer;

Au départ, nous disposions d'un réseau non-édité comprenant l'agglomération urbaine de l'île de Montréal. De là, nous avons extrait un réseau correspondant au secteur à déneiger. Ce secteur comprend à la fois les rues qui doivent être déneigées et les rues qui se situent au contour des précédentes et qui ne servent qu'à passer d'un arc à déneiger à un autre. On peut voir le réseau à la figure 6.1.

Ensuite, nous avons obtenu les informations pertinentes concernant le réseau telles que les sens uniques et les interdictions de virage. Nous avons édité le réseau, i.e. rendu ces informations accessibles en mémoire. A chaque intersection, nous nous sommes également assurés que les connections étaient correctes, puis nous avons fait certaines modifications au réseau lorsque nécessaires, à savoir destruction et création d'arcs. Encore là, ces modifications étaient rendues possibles grâce à l'éditeur de réseau.

5.2 Base de tronçons

Tel que mentionné précédemment, nous voulons travailler avec des tronçons i.e. des regroupements d'arcs. Chacun de ces tronçons est représenté par un nom de rue, un noeud origine et un noeud destination déterminés selon le sens du réseau ainsi que le sens dans lequel on peut le parcourir. De plus, on y indique si c'est un tronçon prioritaire. Chaque tronçon est jumelé à un autre ayant la même représentation et n'étant différencié que par son numéro et le nom du côté de la rue. Par exemple, pour un tronçon orienté "Nord-Sud", on a deux tronçons associés aux côtés de rues "est et ouest". S'il s'agit d'une rue à sens unique, le premier tronçon est le côté droit du sens unique lorsqu'on le parcourt dans le sens permis, tandis que s'il s'agit d'une rue à double sens, le premier tronçon est le côté droit lorsqu'on le parcourt dans le sens du réseau. Cette base est emmagasinée dans le format BTrieve et certaines facilités de l'éditeur de réseau sont incorporées à un programme servant justement à créer cette base. Il est à noter que dans la base, on ne dédouble pas l'information relative à chaque couple de tronçons jumelés, question de sauver de l'espace mémoire.

5.3 Grande table de distances entre les tâches

Etant donné un ensemble de tâches situées sur les arcs du réseau, une table de distances donne la distance entre chacune de celles-ci. Pour faire appel à cette procédure, nous devons d'abord produire un fichier d'entrée contenant les informations sur chacune des tâches. Afin

de pouvoir repérer chacune de ces tâches, on parcourt les arcs composant l'ensemble des tronçons présents dans la base. Pour chacun de ces arcs, dépendant du fait qu'il est à sens unique ou non, on génère les informations d'une tâche ou deux. Les informations contiennent entre autres le numéro de la tâche, le numéro de région, le numéro d'arc et indique si l'arc est dans le sens du réseau. En fait, comme nous devons déjà traiter plusieurs tâches, il n'est pas nécessaire de générer deux tâches pour un sens unique car de toute façon nous obtiendrions deux résultats identiques. Pour ce qui est des arcs à double sens, il est à noter qu'il y a un tâche dans le sens du réseau et une autre qui ne l'est pas.

Pour produire la table de distances entre chacune des tâches présentes dans le fichier d'entrée, nous utilisons une adaptation de l'algorithme de Dijkstra sur arcs tenant compte des permissions de virages, des sens uniques et qui pénalise les virages en U dans les culs-de-sac. Pour chacun des couples de tâches, il s'agit de déterminer la valeur du plus court chemin sachant que la distance est prise entre la fin de la première tâche et le début de la deuxième tâche. Si on avait généré deux tâches pour chacun des arcs à sens unique, on aurait eu un problème de 470 tâches et une table de 441,800 K. Puisqu'on a procédé autrement, le problème ne comporte que 354 tâches mais on obtient tout de même une table de taille imposante soit 250,632K.

5.4 Petite table de distances entre les tronçons

A cause de sa dimension, nous évitons de mettre la table de distances entre chaque couple de tâches en mémoire lors de l'exécution du programme. Nous la remplaçons par une petite table de distances entre chaque couple de tronçons. La dimension est alors réduite à 29,768 K car le nombre de tronçons n'est que de 122. Afin d'initialiser la petite table de distances, il s'agit alors d'extraire les bonnes valeurs de la grande table de distances.

Pour chaque tâche identifiée par son numéro de région et son numéro d'arc dans la grande table de distances, on prend note si elle est dans le sens du réseau ou non. On met dans un tableau toutes les distances calculées précédemment à l'aide de l'adaptation de l'algorithme de Dijkstra et on rend disponible toute l'information concernant les tronçons.

Voici les grandes lignes de la procédure initialisant la petite table de distances:

P1 va du premier au dernier tronçon

P2 va du premier au dernier tronçon

On extrait la distance entre le tronçon origine P1 et le tronçon destination P2 et on la place dans la petite table de distances

Afin d'extraire une distance, il faut déterminer le numéro d'arc et de région du dernier arc du tronçon origine et du premier arc du tronçon destination sachant le sens de la circulation des tronçons origine et destination. De là, découle les numéros des tâches correspondantes et on peut alors extraire directement la distance désirée dans la grande table de distances.

5.5 Voyageur de commerce

Avant de faire appel à la routine de voyageur de commerce qui est une adaptation de l'algorithme de Carpaneto et Toth [4], il y a quelques préparatifs à faire. On doit d'abord déterminer si on doit considérer la présence de dépôts lors de l'appel de la routine du voyageur de commerce. Si on traite la première étape d'une journée, il n'y a pas de dépôt origine représenté par un tronçon mais il peut y avoir un dépôt destination, représenté par le premier tronçon de l'étape suivante s'il est déjà spécifié. Si on traite une étape intermédiaire, il est possible qu'on ait à la fois un dépôt origine et un dépôt destination qui sont respectivement le dernier tronçon de l'étape précédente et le premier tronçon de l'étape suivante s'ils sont déjà spécifiés. Si on traite la dernière étape d'une journée, il peut y avoir un dépôt origine représenté par le dernier tronçon de l'étape précédente s'il est déjà spécifié mais il n'y a pas de dépôt destination représenté par un tronçon. Notons qu'on ne considère pas de dépôt origine pour la première étape ni de dépôt destination pour la

dernière étape car par convention nous avons décidé de ne pas considérer les distances aller-retour du garage vers le territoire à déneiger.

On doit ensuite bâtir une table de distances entre les tronçons de l'étape concernée et les dépôts s'il y a lieu. Si on a "n-1" tronçons dans l'étape, ces "n-1" tronçons se voient assigner un nouveau numéro et on doit alors faire une table de correspondance entre les deux types d'indexation. La table de distances est de dimension "n" par "n" car on doit considérer la présence d'un dépôt qui correspond au $n^{ième}$ sommet. On s'occupe à prime abord d'établir les distances entre les "n-1" tronçons. La distance d'un tronçon au dépôt est égale à zéro, à moins qu'il existe un dépôt destination, auquel cas elle devient égale à la distance à ce dépôt destination. De la même façon, la distance du dépôt à un tronçon est égale à zéro, à moins qu'il existe un dépôt origine, auquel cas elle devient égale à la distance à ce dépôt origine.

On appelle alors la routine du voyageur de commerce en envoyant entre-autres comme paramètres la table de distances nouvellement créée et le nombre de tronçons "n-1". Cette routine s'avère exacte lorsque le nombre de tronçons n'est pas tellement plus élevé que quinze. Pour les problèmes plus gros, elle ne l'est plus non pas parce que c'est une heuristique mais parce qu'on a limité volontairement la taille de l'arbre d'énumération.

Le résultat de l'appel de cette routine est un tableau de pointeurs qui indique dans quel ordre on devra traiter les tronçons pour minimiser le détour encouru. Si l'ordre a changé, il faut modifier l'itinéraire pour l'étape en question, et modifier le temps utilisé pour l'étape concernée et le temps utilisé de l'étape suivante de la même journée si on a changé le dernier tronçon de l'étape.

La routine du voyageur de commerce est écrite en Microsoft Fortran et est appelée de notre programme écrit en Microsoft C. On a pris soin de faire toutes les adaptations nécessaires.

CHAPITRE 6

APPLICATION A UN EXEMPLE REEL ET RESULTATS

Le territoire que nous analysons dans notre exemple est illustré à la figure 6.1. Pour ce territoire, il n'y a qu'une seule équipe de chargement. En effet, si l'accumulation est assez importante pour justifier l'utilisation de deux équipes (souffleuses), on préfère les faire se suivre plutôt que de les affecter à des tronçons différents. Ceci a pour effet de réduire l'allège qui serait accrue autrement du fait qu'il faudrait passer deux fois sur les grandes artères pour arriver à ramasser toute la neige.

Chaque journée est partagée en deux périodes de travail, la première allant de 14h à 18h et la seconde de 19h à 6h. La première période comprend deux étapes, soit celle de 14h à 16h et celle de 16h à 18h, tandis que la deuxième période n'est constituée que d'une seule étape. Les étapes permises et défendues pour placer les différents tronçons sont telles que mentionnées en 4.2.1 et en 4.3.1.

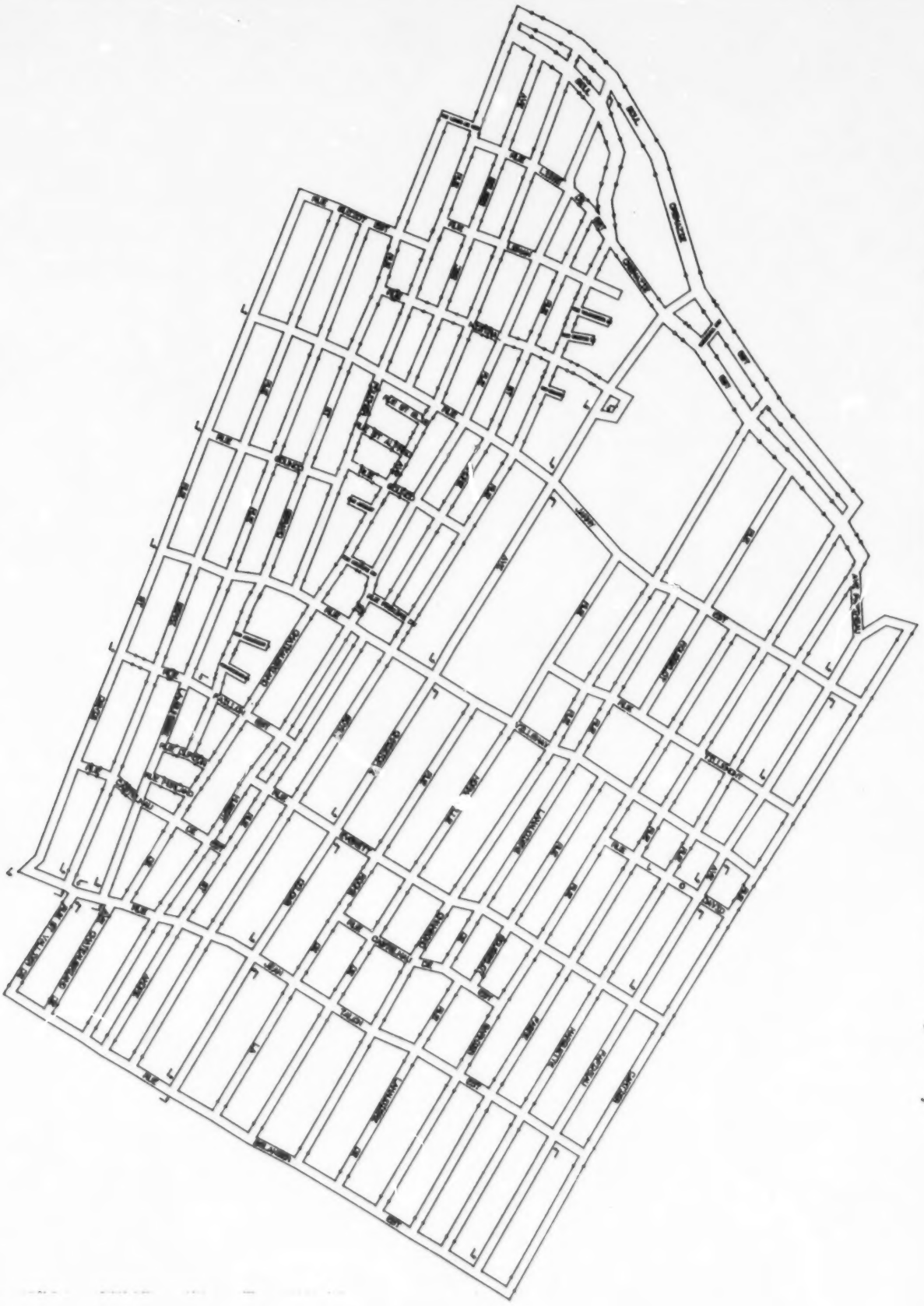

Figure 6.1 Réseau

En ne travaillant que de 14h à 6h, on évite de charger pendant l'heure de pointe matinale et on respecte tout de même la règle qui veut que l'équipe de chargement soit au travail pour un minimum de 16 heures par jour. Il est à noter qu'à l'heure des repas, soit de 18h à 19h et de 24h à 1h, on laisse la souffleuse là où elle est rendue.

Pour ce qui est de la planification des périodes de chargement, on essaie de faire venir la main-d'oeuvre pour des périodes de travail les plus complètes possible. La dernière période peut ne pas être remplie à capacité mais d'un autre côté, elle peut servir à parer aux imprévus. On voit aussi à envoyer au fur et à mesure des tracteurs-chargeurs dans les culs-de-sac pour pouvoir ramasser la neige en passant avec la souffleuse.

6.1 Test avec les données d'une tempête type

Pour faire les tests, nous nous sommes basés sur la tempête qui était la mieux documentée (i.e. la tempête pour laquelle les données concernant les temps de chargement étaient les plus complètes) afin de mieux estimer les vitesses de chargement pour chacun des tronçons. Pour cette tempête dont le début du chargement avait lieu lors d'une période de nuit, nous présentons brièvement les résultats obtenus. D'abord, en ne permettant que l'insertion simple, on a obtenu une allège totale de 33704 m. Ensuite, lorsque nous avons permis à la fois l'insertion simple et l'insertion en parallèle, on a réduit l'allège totale à

31948 m. Il est à noter que la distance d'allège est importante puisqu'elle représente environ le tiers de la distance parcourue étant donné que la longueur totale des arcs à déneiger est de 57848 m.

Vu l'état des données disponibles, il est difficile de comparer ces résultats avec l'allège que comportait les opérations du contracteur pour cette tempête. En effet, il arrivait que les données disponibles indiquaient un temps de chargement pour un ensemble de tronçons sans détails précis indiquant l'ordre selon lequel ils étaient traités. Certaines projections d'allège totale à partir de ces données indiquent toutefois qu'elle est supérieure à celle obtenue avec notre procédure.

On a constaté que les regroupements de tronçons aux diverses périodes appartenaient à des secteurs bien définis du territoire. De plus, ces regroupements se comparent avantageusement à ceux qu'on retrouve dans les opérations du contracteur.

Finalement, même si la méthode a été implantée en langage C sur micro-ordinateur, il importe de noter la rapidité d'exécution. En effet, deux minutes de temps d'exécution sur un IBM-PC-AT ont été suffisantes pour générer une solution pour le territoire illustré à la figure 6.1.

6.2 Difficulté d'obtenir plus de données

Pour arriver à construire un itinéraire de chargement lorsque survient une tempête de neige, il faut être en mesure d'estimer le temps de chargement des différents tronçons en fonction du nombre de centimètres de neige tombée. Malheureusement, le nombre réduit de tempêtes lors de l'hiver 1987-1988 n'est pas vraiment suffisant pour obtenir des estimés robustes. De plus, souvent les données disponibles ne sont pas complètes.

6.3 Améliorations et extensions

Pour bien calibrer le modèle, il est essentiel d'obtenir des données plus complètes. Il est évident qu'une mauvaise planification du déneigement a des impacts relativement importants sur les opérations. En effet, si la planification n'inclue pas suffisamment de tronçons, on doit rechercher d'autres tronçons à déneiger sur lesquels il n'y a pas de voitures garées. Evidemment, dans cette situation, il peut arriver que l'on déneige les deux côtés d'un même tronçon et cela est permis car on n'a pas explicitement interdit le stationnement sur les deux côtés du tronçon en question. Par contre, si la planification inclue trop de tronçons, alors un ou plusieurs de ceux-ci ne peuvent être déneigés malgré l'installation des panneaux interdisant le stationnement pendant cette période. Dans ce cas, ces tronçons doivent être nécessairement déneigés dans les cinq premières heures de la prochaine période.

Par ailleurs, comme on peut être amené à apporter des changements à l'itinéraire généré pour une période donnée non seulement à cause de mauvaises prévisions mais également pour plusieurs autres raisons (bris mécaniques, par exemple), on se servira du programme comme d'un outil quotidien. Ainsi, lorsque des changements surviendront, on fera une nouvelle exécution à partir des nouvelles informations dont on dispose.

Afin de permettre à l'usager d'intervenir lors de la création d'un itinéraire, il reste à compléter un menu interactif. Cela permettrait par exemple à l'usager de choisir les germes pour certaines étapes, de placer un tronçon à un endroit spécifique dans une étape, ou d'interdire la sélection d'un tronçon à une étape donnée. De plus, on pourrait intégrer dans le programme des situations particulières : par exemple au cours de la fin de semaine, on peut sélectionnner des tronçons prioritaires autant de premier ordre que de deuxième ordre. Finalement, on pourrait adapter le programme à certaines exigences particulières de certains contracteurs : par exemple, certains ne permettent pas de déneiger un tronçon secondaire entre deux prioritaires dans une même période, d'autres préfèrent déneiger au moins un côté de chacun des tronçons secondaires avant de déneiger le deuxième côté de l'un d'eux. Plus on pourra intégrer les différentes exigences particulières des utilisateurs éventuels, meilleures seront les chances que ce programme soit utilisé.

CONCLUSION

Au cours de ce mémoire, nous avons réussi à traiter un nouveau problème de recouvrement d'arcs se prêtant tout aussi bien à une formulation en terme de recouvrement de noeuds. Il s'agit de générer les itinéraires d'enlèvement de la neige dans un secteur urbain sachant que chacun des deux côtés des rues doivent être inclus dans les itinéraires générés. Nous devons nous assurer que les rues prioritaires sont déneigées dans les plus brefs délais possibles tout en respectant les diverses contraintes. Rappelons qu'on a un ensemble de journées pour procéder au déneigement de l'ensemble du territoire et qu'il n'est pas permis de déneiger les deux côtés d'une même rue au cours de la même demi-journée.

Chaque segment de rue devant être déneigé est vu comme une tâche. L'objectif est alors d'agencer ces tâches tout en minimisant la distance totale d'allège encourue. On fait face à un problème de grande taille et on le traite avec une méthode heuristique plutôt qu'avec une méthode exacte. Pour réduire la taille du problème, on a choisi de considérer comme unité de base, les tronçons qui correspondent à des regroupements de tâches. En résolvant le problème d'enlèvement de la

neige par une heuristique constituée de deux variantes de la méthode d'insertion, nous avons obtenu des résultats très satisfaisants. On place d'abord les tronçons prioritaires en insertion simple au cours des premières périodes. Par la suite, on comble les étapes de ces périodes avec des tronçons secondaires. Finalement, pour les tronçons qui restent, on a le choix d'utiliser uniquement de l'insertion simple ou bien une combinaison d'insertion en parallèle et d'insertion simple. Au moment d'insérer un tronçon dans une étape donnée, on voit à insérer celui qui occasionne le plus petit détour ou, le cas échéant, celui qui est le plus long. Pour arriver à réaliser ce projet, nous nous sommes également servis de plusieurs programmes existants au Centre de Recherche sur les Transports, notamment un éditeur de réseau et des algorithmes pour les problèmes de plus court chemin et de voyageur de commerce. Enfin, nous avons obtenu des résultats très encourageants indiquant bien que la méthode employée s'avère un outil intéressant dans la planification des périodes de travail d'une équipe de chargement.

BIBLIOGRAPHIE

[1] **Beltrami, E.J., Bodin, L.D.** *"Networks and Vehicle Routing for Municipal Waste Collection"*, Networks, Vol. 4, pp. 65-94, 1974.

[2] **Bodin, L. et al.** *"Routing and Scheduling of vehicles and crews : the state of the art"*, Comput. & Ops Res., Vol. 10, pp. 63-211, 1983.

[3] **Bodin, L.D. et Kursh, S.J.** *"A detailed description of a computer system for the routing and scheduling of streets sweepers"*, Comput. & Ops Res., Vol. 6, pp. 181-198, 1979.

[4] **Carpaneto G., Toth P.** *"Some New Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem"*, Management Science, Vol. 26, pp. 736-743, 1980

[5] **Chapleau, L., Ferland, J.A., Lapalme, G. et Rousseau, J.-M.** *"A parallel insert method for the capacitated arc routing problem"*, Operations research letters, Vol. 3, pp. 95-99, 1984.

[6] **Cormier, M.** *"Description du réseau Hotocars"*, Université de Montréal, Centre de Recherche sur les transports, document de travail, 1986.

[7] *"Déneigement"*, cahier des charges spéciales, 1986-1991 type A, Ville de Montréal, Service des travaux publics.

[8] **Krabbe, Donna.** *"Heuristic routing method : a simple answer to collection productivity"*, Solid Wastes Management, Vol. 22, pp. 68-71, 1979.

[9] **Liebman, J.C. et al.** *"Minimum Cost in Residential Refuse Vehicle Routes"*, Journal of the environmental engineering division, Vol. 101, pp. 399-412, 1975

[10] **McBride, R.** *"Controlling Left and U-Turns in the routing of refuse collection vehicles"*, Comput. & Ops Res., Vol. 9, pp. 145-152, 1982.

[11] **Pearn, W., Assad, A. et Golden, B.** *"Transforming Arc Routing into Node Routing Problems"*, working paper MS/86-006, University of Maryland, 1986.

[12] **Roy, S. et Rousseau, J.-M.** *"The capacitated Canadian Postman Problem"*, Centre de recherche sur les transports, Publication #587, 1988.